ODV – New Developments

Reiner Schlitzer

Alfred Wegener Institute for Polar and Marine Research

1. ODV version 4.6.0

2. ODV API

- ***beta available now***
- ***to be released fall 2013 before mid-term review***

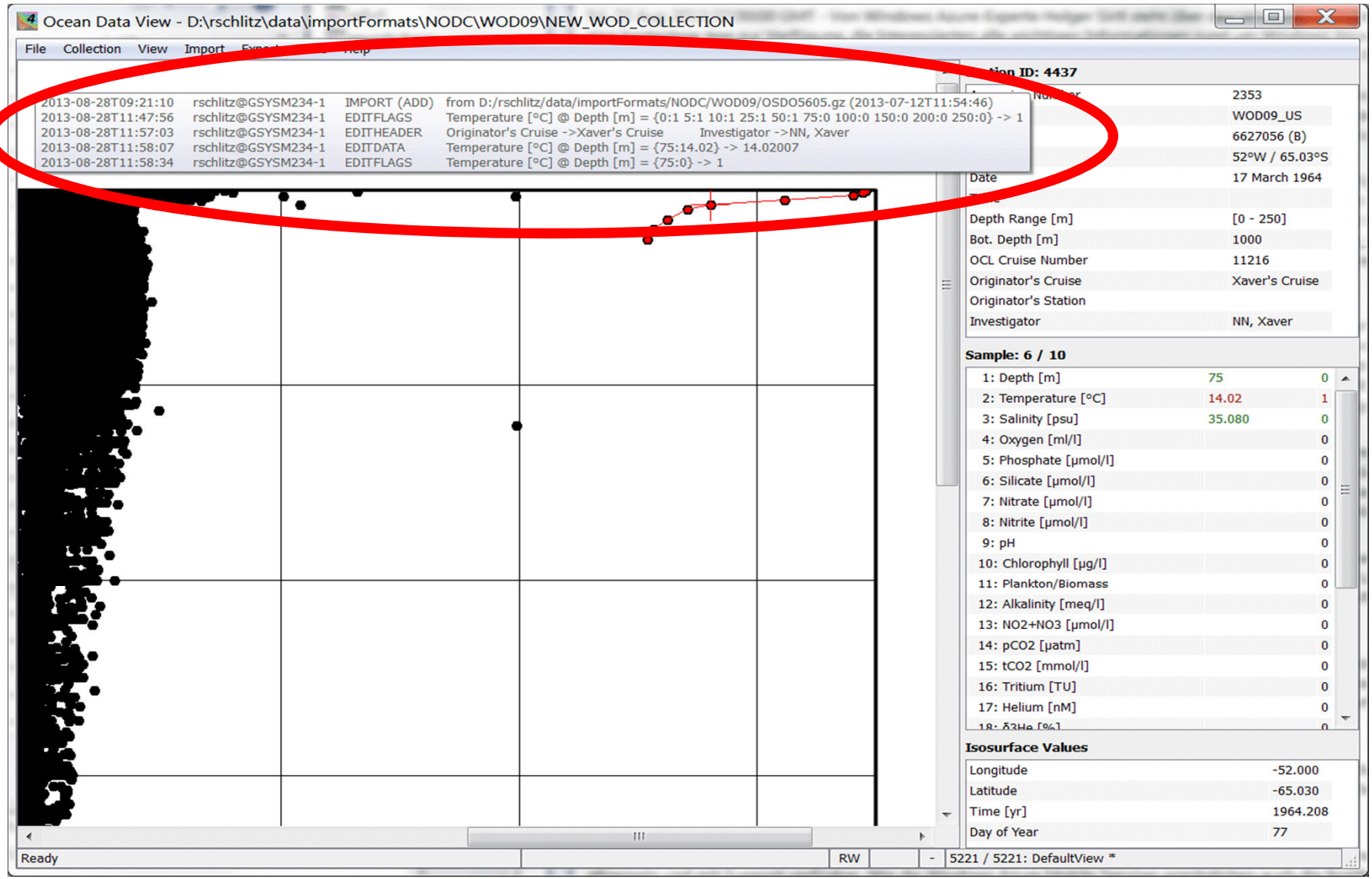
1. ODV version 4.6.0

New ODV collection format ODVCF6

- ***New INDEXED_TEXT data type allowing efficient handling of arbitrary length strings. This permits extended meta variable support***

- ***Per-station data history recording data origin and data edit operations (included when copying or exporting data)***

Per-station data history



Per-station data history in spreadsheet files

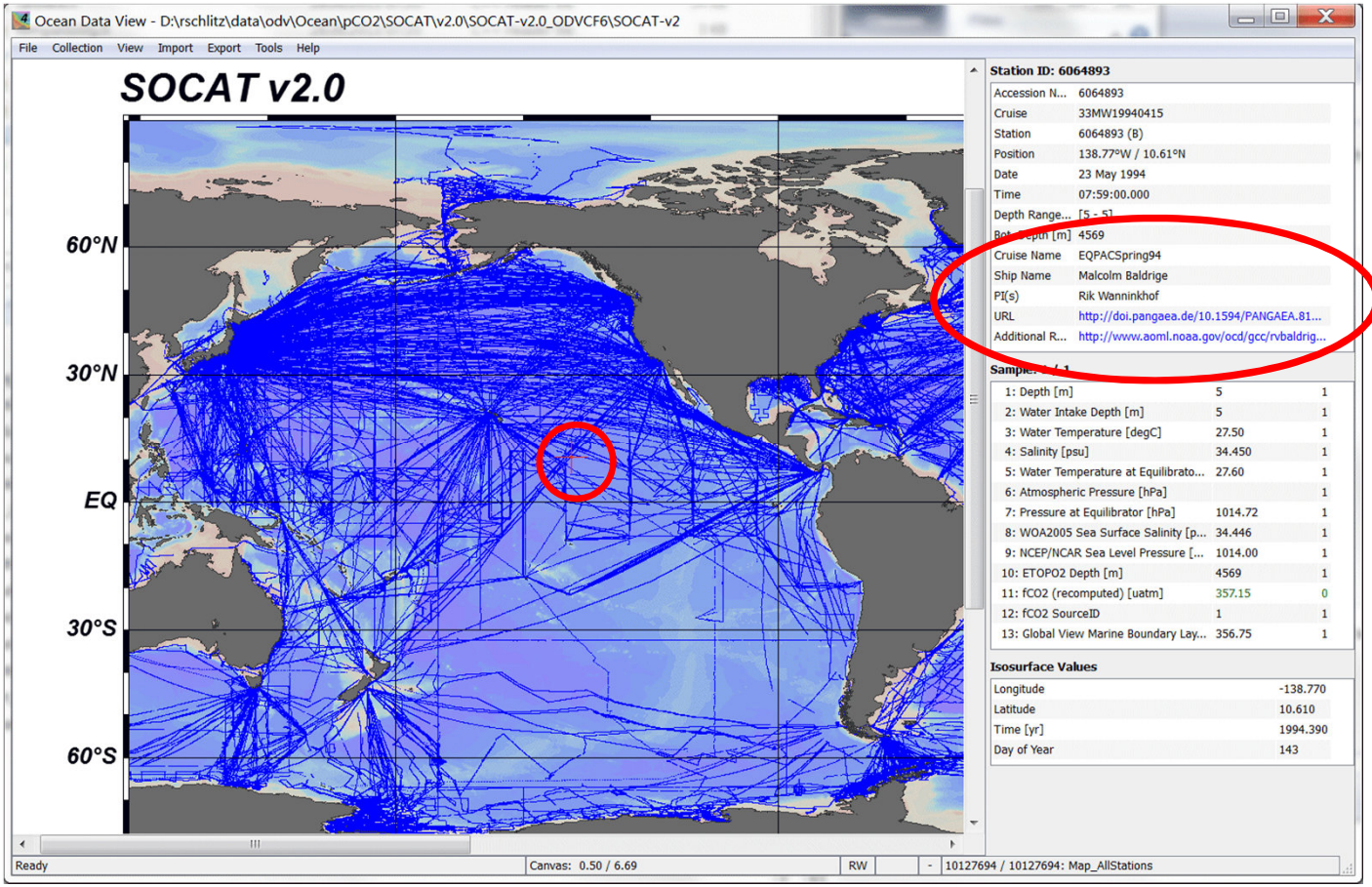
```
                                                                                         1210.00 0       0.2400  0       34.67900
                                                                                         1457.00 0       -0.2200 0       34.65300
                                                                                         1757.00 0       -0.5800 0              0
                                                                                         1784.00 0       -0.5800 0       34.64900
                                                                                         1832.00 0       -0.7400 0       34.63500
                                                                                         1869.00 0       -0.7200 0       34.63900
//<History>2013-08-28T09:21:10  rschlitz@GSYSM234-1     IMPORT (ADD)    from D:/rschlitz/data/importFormats/NODC/WOD09/OSD05704.gz (2(
WOD09_US        530397  B       1968-02-19      317.24200       -72.76700       2044    8037            018B    Xaver3  0.00    0
                                                                                         10.00   0       -1.5700 0       33.61400
                                                                                         24.00   0       -1.2600 0       34.07500
                                                                                         49.00   0       -1.7300 0       34.42400
                                                                                         73.00   0       -1.7800 0       34.42900
                                                                                         98.00   0       -1.8000 0       34.43200
                                                                                         147.00  0       -1.8000 0       34.43800
                                                                                         197.00  0       -1.7700 0       34.44400
                                                                                         297.00  0       -1.6900 0       34.46200
                                                                                         398.00  0       -1.3300 0       34.49400
                                                                                         500.00  0       -0.6100 0       34.56600
                                                                                         600.00  0       0.1100  0       34.63300
                                                                                         702.00  0       0.4200  0       34.66600
                                                                                         807.00  0       0.4600  0       34.67600
//<History>2013-08-28T09:21:10  rschlitz@GSYSM234-1     IMPORT (ADD)    from D:/rschlitz/data/importFormats/NODC/WOD09/OSD05704.gz (2(
//<History>2013-08-28T12:54:07  rschlitz@GSYSM234-1     EDITHEADER      Investigator ->Xaver    </History>
//<History>2013-08-28T13:10:27  rschlitz@GSYSM234-1     EDITHEADER      all Investigator Xaver1->Xaver3 </History>
//<History>2013-09-22T12:16:10  rschlitz@GSYSM234-1     EDITFLAGS       Salinity [psu] @ Depth [m] = {807:0} -> 1</History>
//<History>2013-09-22T12:16:44  rschlitz@GSYSM234-1     EDITDATA        Oxygen [ml/l] @ Depth [m] = {147:7.09} -> 7.090004</History>
//<History>2013-09-22T12:17:15  rschlitz@GSYSM234-1     DELETEDATA      Nitrate [Î¼mol/l] @ Depth [m] = {147:28.4}</History>
```

- ***Data block***
- ***History block***

  //<History> …. </History>

Extended Metadata Support in ODV

Example: *SOCAT v2 pCO_2 Data*

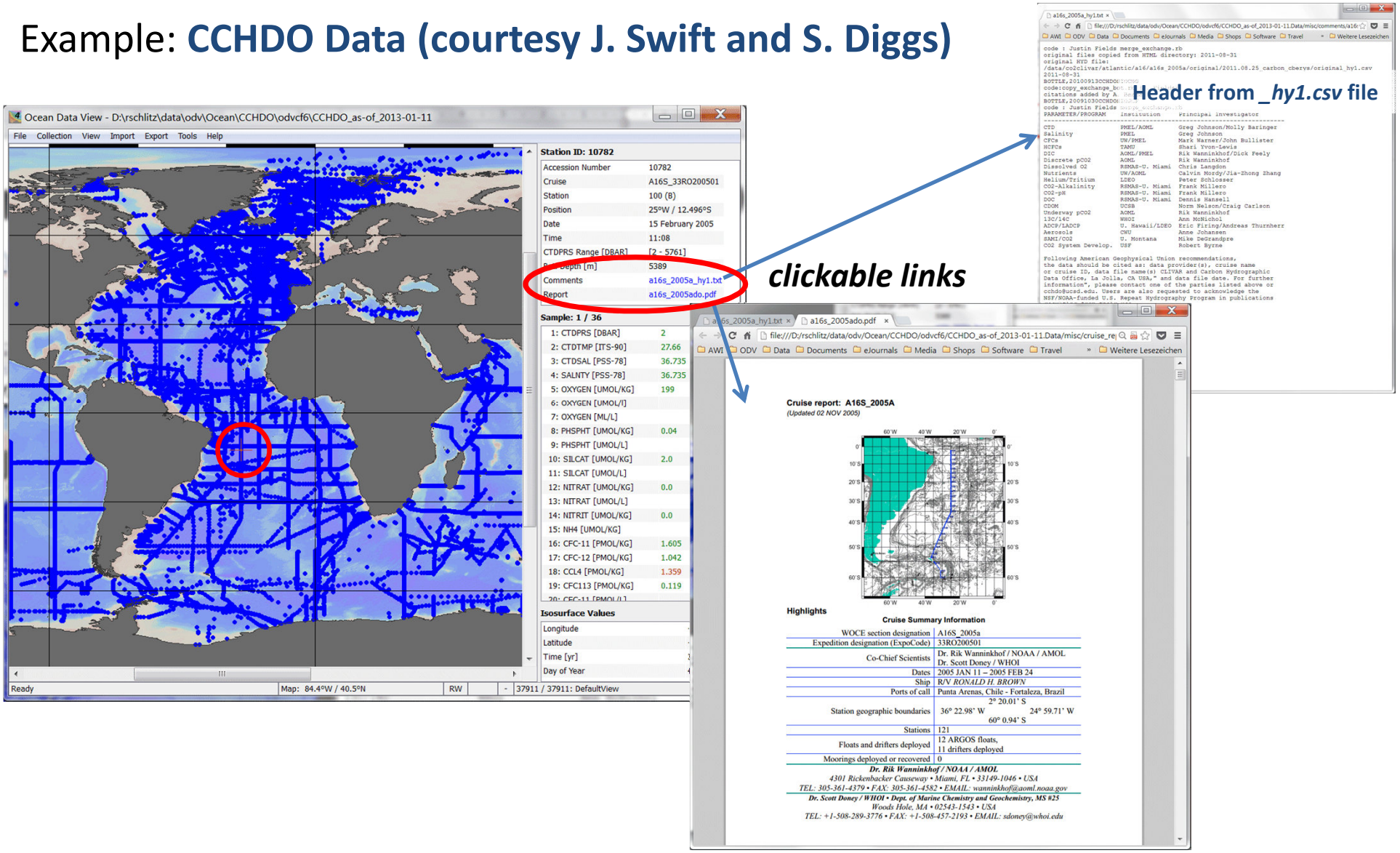


Clicking on any one of the more than 10 million stations instantly shows cruise, ship and PI information.

Clicking on the blue URLs brings up the data description documents in the web browser

Extended Metadata Support in ODV

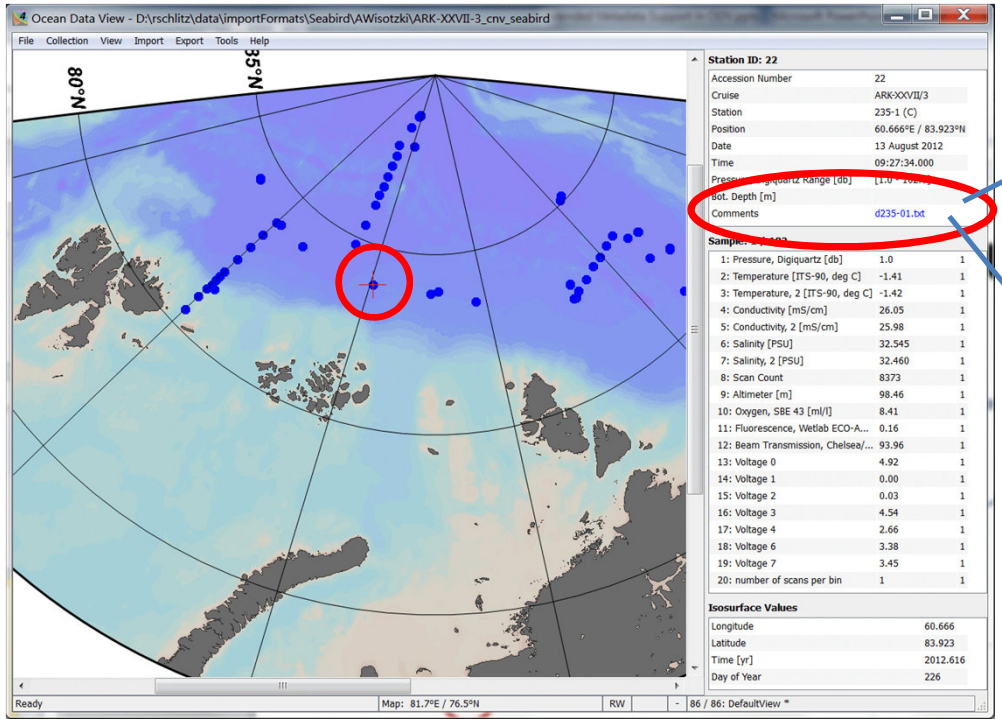
Example: **CCHDO Data (courtesy J. Swift and S. Diggs)**

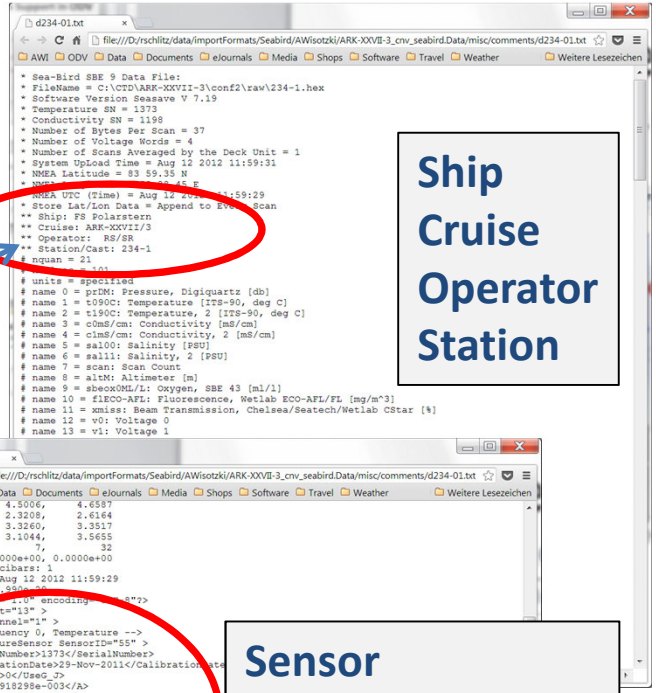


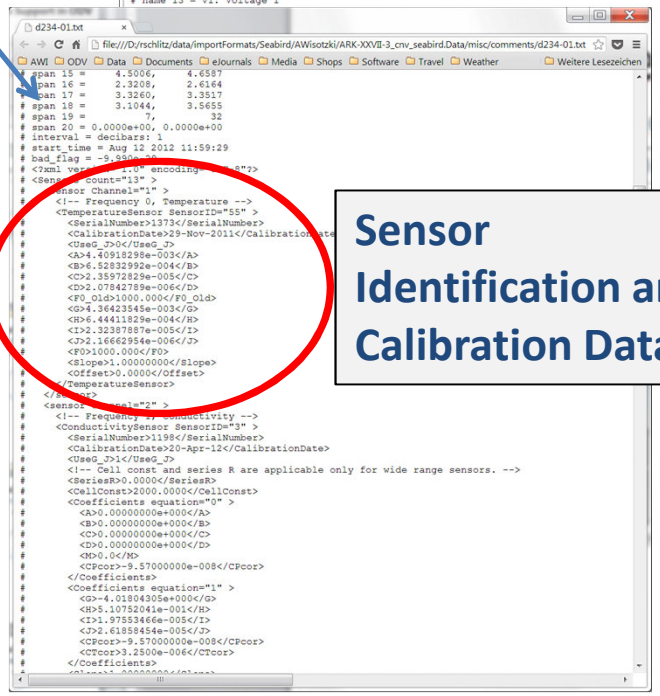
Extended Metadata Support in ODV

Example: *Sea-Bird .cnv Data*

Header from *.cnv* file containing additional metadata

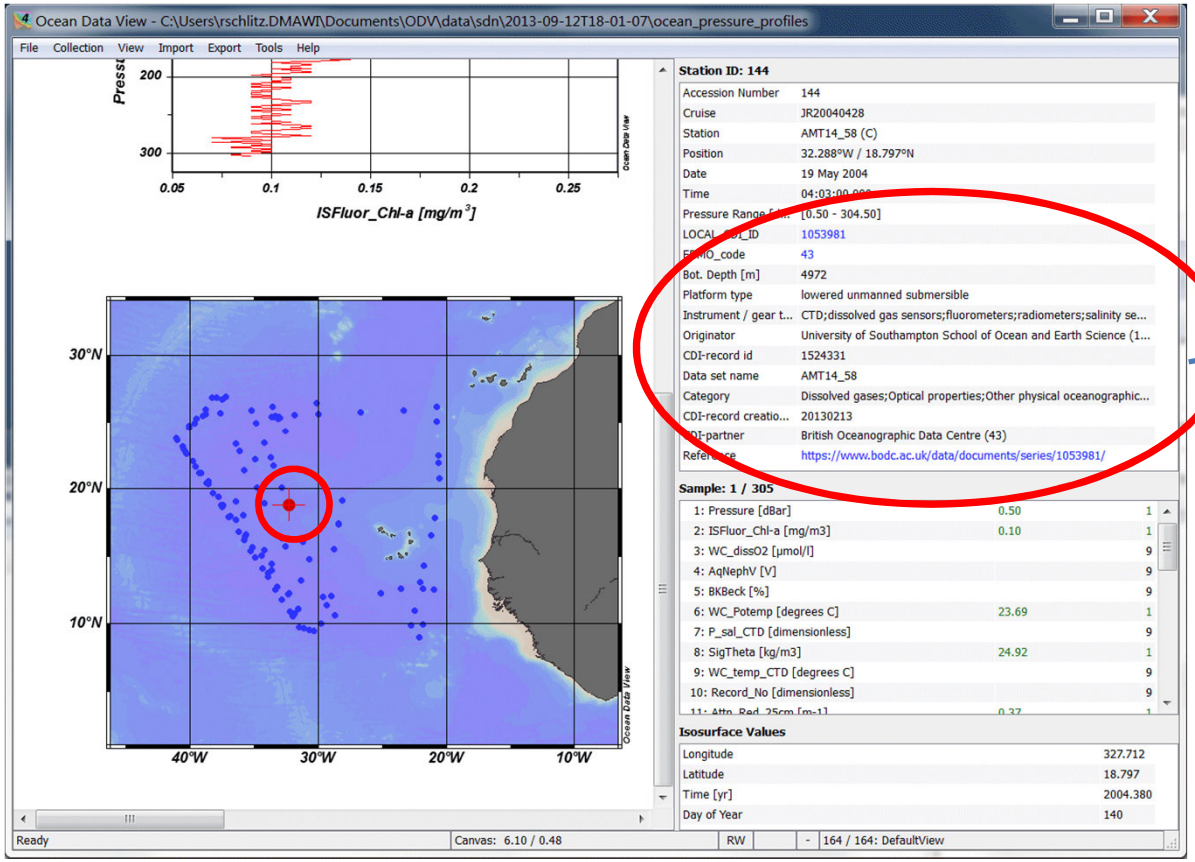






Extended Metadata Support in ODV

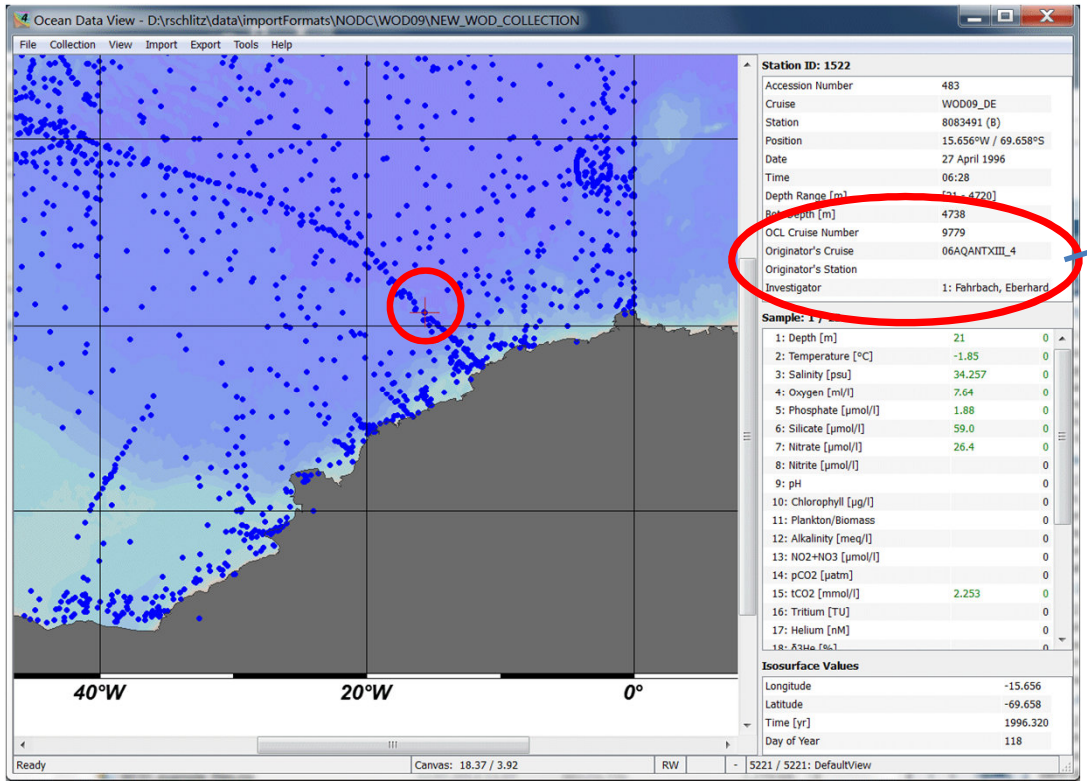
Example: ***SeaDataNet Data***



- LOCAL_CDI_ID (clickable)
- EDMO_code (clickable)
- Platform Type
- Instrument / Gear Type
- Originator
- CDI-Record ID
- Data Set Name
- Category
- CDI-Record Creation Date
- CDI-Partner
- Reference (clickable)

Extended Metadata Support in ODV

Example: ***World Ocean Database***



OCL Cruise Number
Originator's Cruise
Originator's Station
Investigator Names

2. ODV Application Programming Interface (API)

ODV collection format…

- ***Accomodates many data types***
- ***Provides dense storage***
- ***Allows very fast random access***
- ***Is platform independent***

Past:

Data in ODV collections only accessible via Ocean Data View Software.

Future:

Provide ODV Application Programming Interface (API) allowing developers to create their own data access applications for …

custom QC
scientific data processing
web-based data delivery
…

What it consists of:

- ***Set of header files providing the interface (ODV classes and functions; to be included in user code)***
- ***Platform-specific compiled library to link against (Windows: odv4api.dll; Linux: libodv4api.so; Mac OSX: libodv4ap.dylib)***
- ***Set of documentation and example files***
- ***Supported languages: C++ (native), Java***

Features provided:

- *Read access to data, metadata and quality flag values in ODV collections (all formats, including the new ODVCF6)*
- *ReadWrite access will be added in future version*
- *User application code is platform independent (Windows, Mac OS X, Linux, Unix)*

Key Concepts:

ODVCollection

- *Holds values for arbitrary numbers of metadata and data variables for an arbitrary number of stations.*

ODVVariable

- *Represents a metadata or data variable and holds its properties.*

ODVStation

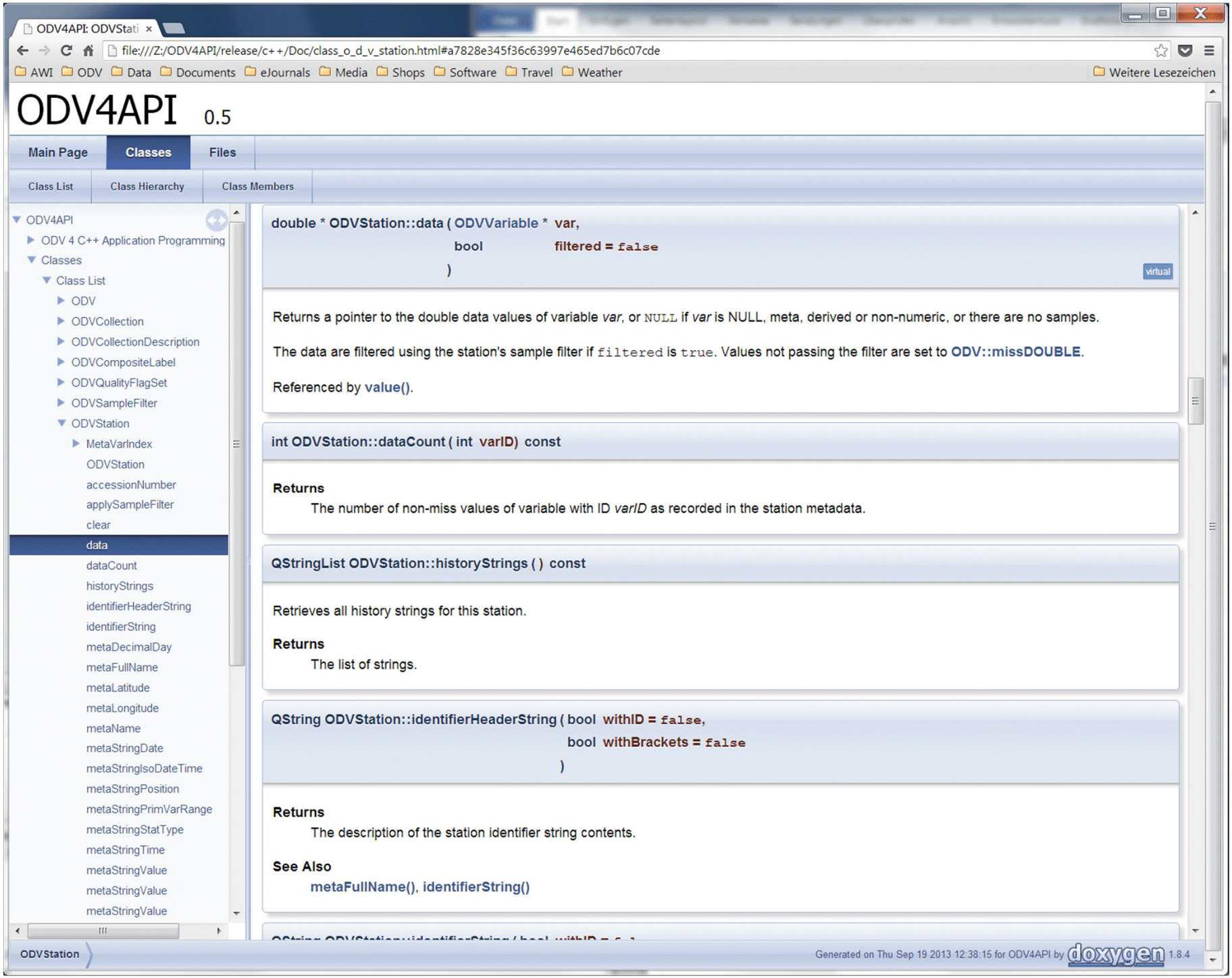
- *Represents sampling event with given space/time coordinates. Holds one value per metadata variable, and data values for every data variable and sample. Also holds quality flags for all metadata and data values.*

C++ Example Code (1/2)

```
/* create an ODVCollection object and open the collection in ReadOnly mode */
ODVCollection collection("c:/odv/test_collection.odv");
ODV::Status status=collection.open(ODV::ReadOnly);


/* retrieve number of metadata and data variables in collection */
int metaVarCount=collection.metaVarCount();
int dataVarCount=collection.basicVarCount();

/* obtain pointer to metadata variable varID (0-based index) */
ODVVariable *var=collection.metaVar(varID);

/* obtain pointer to data variable varID (0-based index) */
ODVVariable *var=collection.var(varID);

/* retrieve number of stations in collection */
int stationCount=collection.stationCount();


/* create an ODVStation object and read data of station statID in
   the collection. note that station IDs used in the readData() call
   are zero-based integers, e.g. 11 for 12th station. */
ODVStation station(&collection);
station.readData(statID);
```

C++ Example Code (2/2)

```
/* retrieve various metadata values */
QString cruiseLabel=station.metaStringValue(ODVStation::MetaCruiseIndex);
QString stationLabel=station.metaStringValue(ODVStation::MetaStationIndex);
double lon=station.metaLongitude();
double lat=station.metaLatitude();

/* retrieve number of samples */
int sampleCount=station.sampleCount();

/* retrieve the data and quality flag values for sample sampleID of
   variable varID. note that sample IDs and variable IDs are 0-based
   integers. */

/* obtain pointer to data variable varID (0-based index) */
ODVVariable *var=collection.var(varID);

/* retrieve data value and quality flag for this variable for sample sampleID */
char qFlag;
double dValue=station.value(var,sampleID,&qFlag);

/* retrieve pointer to data for this variable and access value for
   sample sampleID via this pointer */
double *dPtr=station.data(var);
dValue=dPtr[sampleID];


/* close the collection */
collection.close();
```


ODV4API: ODVStati
file:///Z:/ODV4API/release/c++/Doc/class_o_d_v_station.html#a7828e345f36c63997e465ed7b6c07cde
AWI ODV Data Documents eJournals Media Shops Software Travel Weather
Weitere Lesezeichen
ODV4API 0.5
Main Page
Classes
Files
Class List
Class Hierarchy
Class Members
ODV4API
ODV 4 C++ Application Programming
Classes
Class List
ODV
ODVCollection
ODVCollectionDescription
ODVCompositeLabel
ODVQualityFlagSet
ODVSampleFilter
ODVStation
MetaVarIndex
ODVStation
accessionNumber
applySampleFilter
clear
data
dataCount
historyStrings
identifierHeaderString
identifierString
metaDecimalDay
metaFullName
metaLatitude
metaLongitude
metaName
metaStringDate
metaStringIsoDateTime
metaStringPosition
metaStringPrimVarRange
metaStringStatType
metaStringTime
metaStringValue
metaStringValue
metaStringValue
double * ODVStation::data (ODVVariable * var,
bool filtered = false
)
virtual
Returns a pointer to the double data values of variable var, or NULL if var is NULL, meta, derived or non-numeric, or there are no samples.
The data are filtered using the station's sample filter if filtered is true. Values not passing the filter are set to ODV::missDOUBLE.
Referenced by value().
int ODVStation::dataCount (int varID) const
Returns
The number of non-miss values of variable with ID varID as recorded in the station metadata.
QStringList ODVStation::historyStrings ( ) const
Retrieves all history strings for this station.
Returns
The list of strings.
QString ODVStation::identifierHeaderString (bool withID = false,
bool withBrackets = false
)
Returns
The description of the station identifier string contents.
See Also
metaFullName(), identifierString()
ODVStation
Generated on Thu Sep 19 2013 12:38:15 for ODV4API by doxygen 1.8.4

Availability:

- *C++ and Java test versions available (personal contact or e-mail)*
- *Beta status (tested in-house, Java version tested by IFREMER)*
- *Planned release fall 2013, together with ODV 4.6.0*

Other Languages:

- ***Versions for Perl, PHP, Python, Tcl, Ruby, C#, R, Octave, GO or D can be produced using SWIG wrapper technology.***
- ***We seek cooperation with partners, if support for these additional languages is requested.***